37589

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **COHEN et al.**

Serial No.: 09/651,800 : Group Art Unit: 2178

Filed : August 30, 2000 : Examiner: Kyle R. Stork

For : INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Simona Cohen, Tirtsa Hochberg, Haim Nelken, Ilan Paleiov and Pnina Vortman, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 1-34 therein.

2) Prior to June 29, 2000, we conceived our invention, as described and claimed in the subject application, in Israel, a WTO country. Prior conception of the invention is evidenced by a draft of the patent application sent to us on June 22, 2000, by Dr. Daniel Kligler, of Sanford T. Colb & Co., who was retained by

IBM as outside counsel for the purpose of preparing the present patent application. A copy of the draft, with Dr. Kligler's cover letter, is attached hereto as Appendix A. We note that the claims in this draft are nearly identical to the claims in the application as filed.

3) During July and August, 2000, we continued to work diligently on reduction to practice of our invention. As evidence of this work, we attach hereto copies of internal communications regarding the "Unified Customer Reporting (UCR)" project that were exchanged during this period. These communications, taken from the IBM Lotus Notes Database, are attached hereto as Appendices B-D. Each communication is marked with the date on which it was sent. It was in the context of the UCR project that the invention described in the present patent application was developed. (See, for example, the "Problem Description" on page 1 of Appendix B.)

4) On August 3, 2000, we sent Dr. Kligler our comments on the draft application that he had sent to us on June 29. A copy of the fax cover page to Dr. Kligler and the pages of the draft with our mark-ups is attached hereto as Appendix E.

5) On August 5, 2000, Dr. Kligler sent us the final draft of the application. Dr. Kligler's cover letter to us is attached hereto as Appendix F.

6) Shortly thereafter, we informed Tal Noy-Cohen, IP manager at the IBM Haifa Research Laboratory, that the

revised draft was acceptable. On August 7, 2000, Ms. Noy-Cohen passed the approval on to Dr. Kligler by e-mail. A copy of this e-mail is attached hereto as Appendix G. (The application is referred to in the e-mail by Dr. Kligler's docket number, 38070.)

9) On August 9, 2000, Dr. Kligler's firm sent the patent application to Ms. Noy-Cohen together with documents for us to execute before filing. A copy of the cover letter under which the documents were sent is attached hereto as Appendix H.

10) Because of delays in obtaining the signatures of all the inventors during the summer vacation, Ms. Noy-Cohen instructed Dr. Kligler's firm to file the application without documents. The application was then sent to the United States, where it was filed on August 30, 2000.

We hereby declare that all statements made herein of our knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

סימונה כהן — 5/7/06

Simona Cohen, Citizen of Israel — Date

23 Shimkin Street
Haifa 34750, Israel

תרצה הוכברג	12/07/2006
Tirtsa Hochberg, Citizen of Israel	Date

8 Maabarot Street
Haifa 34461, Israel

	04/07/2006
Haim Nelken, Citizen of Israel	Date

12 Vitkin Street
Haifa 34756, Israel

Ilan Paleiov, Citizen of Israel	Date

P.O. Box 732
Kfar Vradim 25147, Israel

	13/07/2006
Pnina Vortman, Citizen of Israel	Date

21 Netiv Ofakim
Haifa 34467, Israel

37589

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Application of : **COHEN et al.**	:	
Serial No.: 09/651,800	:	Group Art Unit: 2178
	:	
Filed : August 30, 2000	:	Examiner: Kyle R. Stork
	:	
For : INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE		

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

DECLARATION UNDER 37 CFR 1.131

Sir:

We, the undersigned, Simona Cohen, Tirtsa Hochberg, Haim Nelken, Ilan Paleiov and Pnina Vortman, hereby declare as follows:

1) We are the Applicants in the patent application identified above, and are the inventors of the subject matter described and claimed in claims 1-34 therein.

2) Prior to June 29, 2000, we conceived our invention, as described and claimed in the subject application, in Israel, a WTC country. Prior conception of the invention is evidenced by a draft of the patent application sent to us on June 22, 2000, by Dr. Daniel Kligler, of Sanford T. Colb & Co., who was retained by

IBM as outside counsel for the purpose of preparing the present patent application. A copy of the draft, with Dr. Kligler's cover letter, is attached hereto as Appendix A. We note that the claims in this draft are nearly identical to the claims in the application as filed.

3) During July and August, 2000, we continued to work diligently on reduction to practice of our invention. As evidence of this work, we attach hereto copies of internal communications regarding the "Unified Customer Reporting (UCR)" project that were exchanged during this period. These communications, taken from the IBM Lotus Notes Database, are attached hereto as Appendices B-D. Each communication is marked with the date on which it was sent. It was in the context of the UCR project that the invention described in the present patent application was developed. (See, for example, the "Problem Description" on page 1 of Appendix B.)

4) On August 3, 2000, we sent Dr. Kligler our comments on the draft application that he had sent to us on June 29. A copy of the fax cover page to Dr. Kligler and the pages of the draft with our mark-ups is attached hereto as Appendix E.

5) On August 5, 2000, Dr. Kligler sent us the final draft of the application. Dr. Kligler's cover letter to us is attached hereto as Appendix F.

6) Shortly thereafter, we informed Tal Noy-Cohen, IP manager at the IBM Haifa Research Laboratory, that the

revised draft was acceptable. On August 7, 2000, Ms. Noy-Cohen passed the approval on to Dr. Kligler by e-mail. A copy of this e-mail is attached hereto as Appendix G. (The application is referred to in the e-mail by Dr. Kligler's docket number, 38070.)

9) On August 9, 2000, Dr. Kligler's firm sent the patent application to Ms. Noy-Cohen together with documents for us to execute before filing. A copy of the cover letter under which the documents were sent is attached hereto as Appendix H.

10) Because of delays in obtaining the signatures of all the inventors during the summer vacation, Ms. Noy-Cohen instructed Dr. Kligler's firm to file the application without documents. The application was then sent to the United States, where it was filed on August 30, 2000.

We hereby declare that all statements made herein of our knowledge are true and that all statements made on information and conjecture are thought to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application of any patent issued thereon.

_____________________________ _______________

Simona Cohen, Citizen of Israel Date

US 09/651,800

Declaration under 37 C.F.R 1.131 by Cohen et al.

23 Shimkin Street

Haifa 34750, Israel

_______________________ _______________

Tirtsa Hochberg, Citizen of Israel Date

8 Maabarot Street

Haifa 34461, Israel

_______________________ _______________

Haim Nelken, Citizen of Israel Date

12 Vitkin Street

Haifa 34756, Israel

_______________________ 12/7/06

Ilan Paleiov, Citizen of Israel Date

P.O. Box 732

Kfar Vradim 25147, Israel

_______________________ _______________

Pnina Vortman, Citizen of Israel Date

21 Netiv Ofakim

Haifa 34467, Israel

APPENDIX A

Sanford T. Colb & Co.
Intellectual Property Law

FAXED

Beit Amot Mishpat
8 Shaul Hamelech Blvd.
Tel-Aviv 64733, Israel
Tel. 972-3-693-8560

4 Shaar Hagai
P.O. Box 2273
Rehovot 76122, Israel
Tel. 972-8-945-5122

Beit Lev Hagivah
11 Beit Hadfus
Jerusalem 95483, Israel
Tel. 972-2-651-9453

Facsimile: 972-8-945-4556 972-8-949-1040 ◆ e-mail: colbpat@stc.co.il

IBM CONFIDENTIAL
24 pages via fax to 04-8550070

June 22, 2000

Ms. Simona Cohen
IBM ISRAEL
Haifa Research Laboratory
MATAM, Haifa 31905

Re: New U.S. patent application
INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE
Your ref. IL9-2000-0027, our ref. 38070

Dear Simona:

Attached please find a first draft of the above-referenced patent application.

Please review this draft, together with your co-inventors, and let me have your corrections and comments at your earliest opportunity. Note a number of questions that I have marked in **boldface** in the text.

Sincerely yours,

Daniel Kligler, Ph.D.
Sanford T. Colb & Co.

encl.

cc: Adv. Tal Noy-Cohen

INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE

FIELD OF THE INVENTION

The present invention relates generally to information systems, and specifically to methods and systems for integration of heterogeneous data from distributed sources.

BACKGROUND OF THE INVENTION

In today's business environment, many applications need to use data that are warehoused in diverse data sources and repositories. These data are typically expressed in different formats and languages, and are retrieved using different access methods and delivery vehicles. Intermittently, new data sources may be added to the corpus that the application must handle, while existing data sources may be removed or changed. These problems are particularly prevalent in organizations in which databases and applications have developed gradually over the course of years in response to growing Information Systems (IS) demands.

There is therefore a need for tools that enable database information from diverse sources to be integrated into an accessible whole. Such tools are required, for example, in such application areas as customer relationship management, personnel data warehouses, and performance analysis of computer systems and networks. The conventional approach to meeting this need is to create a new data warehouse and to copy into it the required data from the original sources. For example, IBM Corporation (Armonk, New York) offers a product known as "DB2 DataJoiner" that is based on this

sort of approach. DataJoiner is described at www-4.ibm.com/software/data/datajoiner. A solution of this type is also described in U.S. Patent 5,884,310, whose disclosure is incorporated herein by reference.

Another method for manipulating heterogeneous data is described in U.S. Patent 5,345,586, whose disclosure is likewise incorporated herein by reference. A global data directory is provided, which maps the location of data, along with specific data entry attributes and data source parameters. Various tables are used for dealing with the diverse data properties, including an attribute table, a domain table, a routing table and a cross-reference table. The tables are used in accessing the data, in order to provide a system user with a consistent interface to multiple distributed, heterogeneous data sources.

Markup languages are well known in the programming art. The most popular markup language is the Hypertext Markup Language (HTML), which is commonly used on World Wide Web pages and in other document applications. HTML is derived from the Standard Generalized Markup Language (SGML), and uses tags to identify certain data elements and attributes. HTML, however, is not extensible, in the sense that it uses a closed set of tags, and it has little or no semantic structure. In order to address these and other shortcomings, Extensible Markup Language (XML) has more recently been introduced by the World Wide Web Consortium (W3C). XML is defined by a standard available at www.w3.org/XML.

XML allows users to define their own sets of tags, depending on their application needs. Each XML document is associated with a Document Type Definition (DTD),

which specifies the elements that can exist in the document and the attributes and hierarchy of the elements. Many different DTDs have already been developed for different applications, such as "performanceML" for computer system performance evaluation, and "CPEX" for customer relationship management. XML.ORG maintains a registry of available DTDs at xml.org/xmlorg_registry/index.shtml. XML-schema are under development as an alternative to DTDs, as described at www.w3.org/TR/xmlschema-0.

Style languages are used to control how the data contained in a markup language document are structured, formatted and presented. For example, W3C has introduced the Extensible Style Language (XSL) for use in defining style sheets for XML documents. An XSL style sheet is a collection of rules, known as templates. When the rules are applied to an input XML file by a processor running an XSL engine, they generate as output some or all of the content of the XML file in a form that is specified by the rules. (In fact, an XSL style sheet is itself a type of XML document.) XSL includes a transformation language, XSLT, which is defined by a standard available at www.w3.org/TR/xslt. Rules written in XSLT specify how one XML document is to be transformed into another XML document. The transformed document may use the same markup tags and DTD as the original document, or it may have a different set of tags, such as HTML tags. Other style languages are also known in the art, such as the Document Style, Semantics and Specification Language (DSSSL), which is commonly used in conjunction with SGML.

SUMMARY OF THE INVENTION

In preferred embodiments of the present invention, a data integration system provides unified access to data residing in diverse, heterogeneous data sources. The data integration is achieved by mapping all of the data, from all of the diverse sources, to a unified schema defined in a markup language. Preferably, the language comprises XML, and the schema comprises a DTD defined for the particular application domain to which the data belong. Alternatively, other markup languages and other schema may be used for this purpose.

In some preferred embodiments of the present invention, the system comprises an administrator application and a middleware-level lookup engine. The administrator is used to map all relevant fields in the diverse data sources to appropriate elements of the chosen schema. The mappings are stored in a repository, and are then used by the lookup engine to transform the data from the diverse data source to a unified format in the markup language, in compliance with the schema. Database access applications, such as queries, interrogate the diverse data sources through the middleware lookup engine, and so receive responses in the unified markup language format, regardless of the source of the data. As a result, differences in source format and complexities in accessing the diverse data sources are completely transparent to the application.

Preferably, a different unified schema is defined for each different domain in which the data integration system is to be used. The schema in each case should cover all of the types of data that may be relevant to the domain. XML and related markup languages are

advantageous in this regard, since they allow a data hierarchy to be defined, with optional parts that can be omitted or added within a given XML document. Within the defined schema, existing data sources may be changed, and new data sources of substantially any type may be added, simply by adding the required mappings to the repository, without modification to the overall system. These changes can even be made dynamically, while the data integration system is running, without affecting existing applications that access the system. By contrast, data integration systems known in the art are limited to static mappings, and typically require system-level modifications when a data source definition is added, deleted or changed.

{Claim summary will be inserted here in the final version.}

The present invention will be more fully understood from the following detailed description of the preferred embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic, pictorial illustration of a system for integrating diverse data sources, in accordance with a preferred embodiment of the present invention;

Fig. 2 is a block diagram that schematically illustrates functional elements of the system of Fig. 1, in accordance with a preferred embodiment of the present invention;

Fig. 3 is a flow chart that schematically illustrates a method for integrating diverse data sources, in accordance with a preferred embodiment of the present invention; and

Fig. 4 is a schematic representation of a computer display used in generating mappings from diverse data sources to a unified schema.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 is a schematic, pictorial illustration of a system 20 for integrating data from diverse data sources, in accordance with a preferred embodiment of the present invention. The data sources typically comprise databases stored on distributed storage devices 26. System 20 is capable of working with substantially any type of structured data, however, and it is not necessary that the data sources comprise relational databases. A data integration server 24 accesses the data on storage devices 26 and provides the data to a client computer 22, typically in response to a query from the client. The client computer may be in close proximity to the server, or alternatively, it may access the server via a network, such as the Internet. Further alternatively, the functions of the server and the client may be integrated and carried out on a single machine.

Client computer 22 and server 24 preferably comprise general-purpose computer processors, which are controlled by appropriate software to carry out the functions described hereinbelow. The software may be provided to the client and the server in electronic form, by download over a network, for example, or alternatively, it may be furnished on tangible media, such as CD-ROM.

Fig. 2 is a block diagram that schematically illustrates functional elements of system 20, in accordance with a preferred embodiment of the present invention. A middleware layer 30 running on server 24 is responsible for integrating data from diverse sources 32 using XML. For each of data sources 32, an administrator application 34 is used to define mappings from the data source to a unified schema 38, as described in detail

hereinbelow. The schema is preferably specified by a DTD that is defined for the domain to which data sources 32 belong, such as the performanceML or CPEX DTD noted above. The mappings defined by administrator 34 are stored in a repository 36.

A lookup engine 40 uses the mappings in repository 36 to access data sources 32 and to map the diverse data from these sources to unified data 42. The unified data are preferably represented as XML code, complying with schema 38. An application such as a query engine 44, typically running on client 22, is able to access unified data 42 substantially without regard to the differences in format and access methods among sources 32. From the point of view of the application, the diverse sources are all a single body of XML data. Query engine 44 preferably comprises an XML-based query engine, written in the XQL or XML-QL query language, for example. **{Can you provide a reference describing these query languages or query engines?}**

Fig. 3 is a flow chart that schematically illustrates a method by which system 20 integrates the data from sources 32, in accordance with a preferred embodiment of the present invention. At a schema creation step 50, unified schema 38 for the selected domain is specified. Preferably, an existing DTD is selected. Alternatively, a new DTD or other schema may be created, or another type of schema may be used, such as an XML-schema, as mentioned above. The schema should be adequate to cover all of the existing types of data in the domain and, preferably, should be easily extensible

to allow the addition of tags defining new data elements within its hierarchy.

At an access step 52, administrator 34 defines how each of data sources 32 is to be accessed. For example, if the data source is a database, the administrator preferably defines a host name and port (on a network linking server 24 to the appropriate storage device 26), the database name, and a username and password. The administrator then creates mappings from data sources 32 to schema 38, at a mapping step 54. Preferably, the mappings are created using an on-screen editor, as illustrated in Fig. 4. The editor preferably creates XSLT rules, which are used subsequently to carry out the actual mapping. Alternatively, a user of administrator 34 may create the XSLT rules (or other mapping function) by coding it directly, without the aid of a visual editor.

Preferably, each of the mappings created at step 52 is a triplet of the form <source, target, conversion function>. The source is a field or a set of fields in data source 32. The target is an element or an attribute in unified schema 38. The conversion function is a function that is applied to the data in the source in order to create a data value for the target. For example, assuming that system 20 is assembling computer performance data using the performanceML DTD, one of the triplets might be as follows:

- Source - Day.cpu_utilization (the CPU utilization field in the computer's Day table).

- Target - PERFORMANCE.Server.Server_Performance_Info.CPU_utilization (an element in the DTD hierarchy).
- Conversion - floatToPercentage.

Lookup engine 40 creates unified data 42 from data sources 32, at a unified data generation step 56. For each mapping, the lookup engine fetches the appropriate source data from the data sources, transforms the data to XML format, and then maps the data to the target. The mapping is preferably carried out by invoking an appropriate XSL engine, as is known in the art, to operate on the XML source data using the XSLT rules created at step 54. The unified data are then available to query engine 44, at a query step 58. Typically, unified data 42 are not held as a static database, but are rather created dynamically by lookup engine 40 when required by a particular query.

A new data source 32 may be added to system 20, or an existing source may be modified or deleted, at a data source addition step 60. The change in the data sources does not substantively affect unified schema 38 itself. Therefore, it is necessary only to update access information and mappings of the new or modified data source, at steps 52 and 54. Since the schema is unchanged, there is also no need to modify query engine 44 or other applications that access unified data 42.

Fig. 4 is a schematic representation of a computer screen 65 associated with administrator application 34, in accordance with a preferred embodiment of the present invention. Screen 65 is typically displayed on a monitor of client 22, for use in interactively mapping data

sources 32 to schema 38 at step 54 of the method described above. The data sources are identified in a data source window 70, while the mapping targets from the DTD tree or other schema are shown in a DTD window 72. In the example shown in this figure, the selected data source is a cpu_0 field 80 in the perform.c_day table, while the target is a cpu_0 element 82 in the DTD. The selected conversion function, chosen from a function menu, is an intToPercent function 74. Once the user has indicated the chosen source, target and conversion function, he or she selects an add button 76 to enter the mapping in repository 36. A mapping window 78 lists all of the mappings that administrator 34 has created.

Table I lists different types of mappings that may be created by administrator 34 at step 54 to convert source data to target data. These mappings are described here by way of example, and other conversion functions will be apparent to those skilled in the art.

TABLE I - MAPPING TYPES

1. Direct copy from a column in the data source to an element or attribute in the DTD.
2. Apply a conversion function to a column in the data source, and create an element or an attribute in the DTD.
3. Apply a conversion function to a set of columns in the data sources, and create an element or an attribute in the DTD. The columns may belong to different data sources.
4. Select certain rows in the data source, and copy each one to an element or attribute in the DTD.

5. Select certain rows in the data source, apply a conversion function to each selected row, and create an element or attribute in the DTD.
6. Join tables in the data sources to one element in the DTD. The tables may belong to different data sources. **{What does it mean to "join tables"?}**
7. Aggregate data from the data source with a simple function - for example, find the average response time per day. Copy the aggregated data to an element or attribute in the DTD.
8. Aggregate data from the data source with a complex conversion function. Copy the aggregated data to an element or attribute in the DTD. **{What is the definition of "complex" in this context, as opposed to "simple"?}**
9. Mappings that include parameters - for example, copy column cpu_utilization_objective to $cpu_objective, wherein the parameter cpu_objective can get different values in each execution of lookup engine 40.

As noted above, the mappings created at step 54 are preferably recorded as XSLT rules. Tables II and III below are examples of XSLT code that implements two rules of this sort. Table II is a mapping of the first type (direct copy) listed in Table I, while Table III is a mapping of the fifth type (row conversion).

TABLE II – XSLT DIRECT COPY

source - 'fqhn' column of table 'perform.server'
target - attribute 'Server_id' of element 'Server_Configuration_Info' in DTD

conversion function - none (also called Direct)
XSLT template -

```
<xsl:template match="//TABLE[@name='perform.server']">
   <xsl:for-each select="ROW">
      <Server_Configuration_Info>
         <xsl:attribute name="Server_id">
            <xsl:value-of select="fqhn/@Value"/>
         </xsl:attribute>
      </Server_Configuration_Info>
   </xsl:for-each>
</xsl:template>
```

TABLE III – XSLT ROW CONVERSION

source - all rows of table 'perform.c_day'
target - element 'Server' in DTD
conversion function - 'FillStaticInfo' method of the Java class 'FillStaticServerInfo'
XSLT template -

```
<xsl:template match="*">
   <xsl:variable name="group1"
     select="//TABLE[@name='perform.c_day']/ROW"/>
   <xsl:variable name="answer"
     select="java:FillStaticServerInfo.FillStaticInfo
     ($group1)"/>
   <Server>
     <xsl:value-of select="$answer"/>
   </Server>
</xsl:template>
```

Although preferred embodiments described herein make use of certain particular markup languages and tools, such as XML, DTDs and XSLT, further embodiments of the

present invention using other markup languages and associated tools will be apparent to those skilled in the art. Furthermore, although these preferred embodiments relate particularly to methods for reading data from data sources 32, the principles of the present invention may similarly be extended, *mutatis mutandis*, to carry out database transactions, such as writing data. It will thus be appreciated that the preferred embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and subcombinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art.

CLAIMS

1. A method for processing source data from a plurality of diverse sources in a selected data domain, comprising:

specifying a unified schema that lists markup tags in the selected data domain that can exist in a document in the markup language;

defining correspondences of data fields from the sources to the markup tags listed by the schema; and

mapping the source data in accordance with the correspondences to generate unified data in the markup language.

2. A method according to claim 1, wherein the markup language comprises Extensible Markup Language (XML).

3. A method according to claim 2, wherein specifying the unified schema comprises specifying a Document Type Definition (DTD).

4. A method according to claim 2, wherein defining the correspondences comprises defining data transformation rules in Extensible Style Language (XSL).

5. A method according to claim 4, wherein mapping the source data comprises transforming the data using an XSL engine.

6. A method according to claim 1, wherein defining the correspondences comprises selecting one or more of the data fields in the sources to correspond to one of the markup tags in the schema, and determining a conversion function to apply to the one or more data fields.

7. A method according to claim 6, wherein determining the conversion function comprises determining the

function so as to generate a data element indicated by the corresponding one of the markup tags.

8. A method according to claim 6, wherein determining the conversion function comprises determining the function to generate an attribute of the unified data indicated by the corresponding one of the markup tags.

9. A method according to claim 1, wherein at least some of the source data are represented in a language other than the markup language, and wherein mapping the source data comprises transforming the data to the markup language.

10. A method according to claim 1, and comprising querying the sources by addressing a query to the unified data in the markup language.

11. A method according to claim 10, wherein mapping the source data comprises mapping the source data responsive to the query.

12. Apparatus for processing source data from a plurality of diverse sources in a selected data domain, comprising a data integration processor, which is adapted to receive and store a unified schema that lists markup tags in the selected data domain that can exist in a document in the markup language, and further to receive and store definitions of correspondences of data fields from the sources to the markup tags listed by the schema, and to map the source data in accordance with the correspondences to generate unified data in the markup language.

13. Apparatus according to claim 12, wherein the markup language comprises Extensible Markup Language (XML).

14. Apparatus according to claim 13, wherein the unified schema comprises a Document Type Definition (DTD).

15. Apparatus according to claim 13, wherein the definitions of the correspondences comprise data transformation rules in Extensible Style Language (XSL).

16. Apparatus according to claim 15, wherein the processor is adapted to map the source data by transforming the data using an XSL engine.

17. Apparatus according to claim 12, wherein each of the definitions of the correspondences comprise a selection of one or more of the data fields in the sources to correspond to one of the markup tags in the schema, together with a conversion function to be applied by the processor to the one or more source fields.

18. Apparatus according to claim 12, wherein at least some of the source data are represented in a language other than the markup language, and wherein the processor is adapted to transform the data to the markup language.

19. Apparatus according to claim 12, wherein the processor is adapted to receive and respond to a query addressed to the unified data in the markup language.

20. Apparatus according to claim 19, wherein the processor is adapted to map the source data responsive to the query.

21. Apparatus according to claim 12, and comprising a plurality of distributed data storage devices, which hold the diverse data sources, wherein the processor is adapted to retrieve the source data from the distributed devices.

22. A computer software product for processing source data from a plurality of diverse sources in a selected data domain, the product comprising a computer-readable medium in which program instructions are stored, which instructions, when read by a computer, cause the computer to receive a unified schema that lists markup tags in the selected data domain that can exist in a document in the markup language and to receive definitions of correspondences of data fields from the sources to the markup tags listed by the schema, and to map the source data in accordance with the correspondences to generate unified data in the markup language.

23. A product according to claim 22, wherein the markup language comprises Extensible Markup Language (XML).

24. A product according to claim 23, wherein the unified schema comprises a Document Type Definition (DTD).

25. A product according to claim 23, wherein the definitions of the correspondences comprise data transformation rules in Extensible Style Language (XSL).

26. A product according to claim 25, wherein the instructions cause the computer to transform the data using an XSL engine.

27. A product according to claim 22, wherein the instructions further cause the computer to accept and respond to a query addressed to the unified data in the markup language.

28. A product according to claim 27, wherein the product comprises middleware, which causes the computer to map the source data responsive to the query.

29. A product according to claim 28, wherein at least some of the source data are represented in a language other than the markup language, and wherein the middleware causes the computer to transform the data to the markup language.

APPENDIX B

Simona Cohen on 13/07/2000 at 11:48
Category: Presentations
(Embedded image moved to file: pic29336.jpg)

Unified Customer Reporting (UCR) - Proposal for 2001

Summary
Project Title: Unified Customer Reporting (UCR)
Description (1-2 lines):

Following the UCR pilot in 2000 (if successful), we'll develop integration and correlation of SRM and other data sources. There are three potential data sources - SAP metrics data source, Tivoli notes metrics data source, expanded EPP data source. This integration will need development of enhancements to our visualization applet as well as to the DIX (Data Integration via XML) back-end engine.

In addition we suggest two new directions based on the current UCR pilot:

Enhancing UCR for pervasive devices.
Adding predictions for capacity planning using heuristics and mathematics algorithms.

This document does not include funds for the new directions, but we may do so if IGS is interested.

Proposed Research Leader: Simona Cohen
Proposed Global Services Leader: Gary Quesenberry

Problem Description (a paragraph or so):

In today's business environment, many applications need to use data that is warehoused in diverse data sources and repositories. The data is expressed in different formats and languages, retrieved in different access methods, and through different delivery vehicles. Moreover, new data sources may be added, and existing ones removed or changed frequently.

This problem is more and more common in areas where applications and databases were developed gradually over many years to supply increase demand in organizations for Information Systems (IS). Thus, it is critical with organizations that are using IS technology for many years.

SRM (Server Resource Management) is a system performance data source that measures daily resources of over 5000 servers, including metrics such as CPU utilization, percent of memory used, number of users logged in, etc.

IGS would like to integrate SRM with other system performance data sources including:

SAP metrics - measures metrics of SAP applications such as number of users, number of dialogs, SAP transaction code, etc.
Tivoli notes metrics - measures notes metrics taken from Tivoli such as average number of concurrent users who are using Lotus Notes, total MBytes of mail traffic items, etc.
EPP (End-to-end Probe Performance) -- measures the response time of probes such as round-trip e-mail messages, access to specific web pages, etc. The pilot integrates probes for just part of the servers.

In addition there are two new directions of problems related to UCR:

System administrators would like to access system performance reports by pervasive devices.

Capacity planning is a major problem faced by IGS. UCR integrates system performance information from diverse data sources, and this information can be further leveraged to predictions for capacity planning.

Global Services Business Opportunity (a paragraph or so):

Deliver quality services - Global Services and our customers require new system performance reports that include both server resources performance and applications performance. By looking at the different data sources and seeing how the information is related, they can get a better breakdown of the entire situation. UCR supplies such reports in a novel way.

Cost reduction - The pilot will include some generic components in the back-end. This will allow faster integration of diverse data sources in the future, reduce development cost, and increase reuse and flexibility.

Global Services Business Impact (a paragraph or so):

Improved tools for data integration.
Improved tools for customer reporting.
Improved market share in system management services.
Better competitiveness by having the ability to lower cost structure with better tools and more flexibility.

Proposed Research deliverables to Global Services (with impact of each):

An enhanced administrator application to map from data sources to the unified schema DTD.
impact: faster and easier creation of mappings by a visual application.
An enhanced DIX back-end system including a Lookup Engine that integrates the diverse data sources and creates the unified data. Performance and scaleability should be improved.
impact: generic infrastructure to ease data-integration problems.
An enhanced visualization applet that will report and analyze the unified data. Performance and scaleability should be improved.
impact: leverage system management services.
Ongoing evaluation on report requirements and incorporation to our report applet.
impact: leverage the quality of our report.

The deliverables of the two new directions will be specified if IGS is interested.

A Few Key Proposed Milestones:

Q1
Publish UCR technology in a referred conference.
Study the new data sources and establish connection to them.
Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).
DOU document.
Requirements document.

Q2
Design document.
Integrate SAP metrics data source with SRM. This will strive enhancements to the current components as well as development of new conversion functions.

Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).

Q3

Integrate new EPP metrics with SRM. This will strive enhancements to the current components as well as development of new conversion functions.

Add NLS to SRM by adding a new transformation that uses the IBM Automatic Translation Engine (optional).

Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).

Q4

Integrate Tivoli notes metrics with SRM. This will strive enhancements to the current components as well as development of new conversion functions.

Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).

Add documentation and code turn-over.

The milestons of the two new directions will be specified if IGS is interested.

Key Stakeholders:

Future development owner (who will continue with this project once research is finished):

IGS Outsourcing

Probability of success (what risks would prevent deliverables from being provided, and assuming all deliverables are provided what risks would prevent the proposed impact to Global Services from being realized):

Cost of project (per quarter and total):

1Q2000 - 12 PM
2Q2000 - 12 PM
3Q2000 - 12 PM
4Q2000 - 12 PM
Total - 48 PM = 4PY

ROI (Per year after first deliverable for 5 years):

Details

Introduction (describe the problem, technically, as well as from a business perspective including how large the problem is):

During the first quarter of 1999, IGS has posed to IBM research the following problem:

"IGS is obligated to provide different kinds of data/information to different commercial accounts. Examples include measurement data, reports, billing information, trouble ticket data, inventory data, usage data, capacity data, etc. Today each piece of data is provided in different ways to different accounts. The data comes from different sources, is presented in different report formats, using different access methods, through different delivery vehicles, etc. And beyond that, IGS itself has requirements to access similar data on its accounts for internal purposes."

Unified Customer Reporting (UCR) aims to solve the problem by providing a generic infrastructure for data integration of multiple heterogeneous correlated data sources. UCR is open, extensible, modifiable and web-based for maximum client access. It is based on multi-platform standards and IBM strategic infrastructure including WebSphere, XML technology, Servlets, Java. UCR presents the data as a structured text file in XML, so that it can be viewed and used by applications other than UCR applications. XML is rapidly becoming accepted as the standard for data interchange-both across the Web and between applications.

Solution (describe what will be done and how it will solve the problem):

The solution includes a back-end and a front-end with XML data in the interface between them. The back-end includes a novel method and a middleware system for integrating diverse data sources using eXtensible Markup Language (XML). The front-end is based on LifeLines (sometimes called also TimeLines) that was developed by the Watson Research Lab and the Human Computer Interaction Lab at the University of Maryland. We'll describe each part separately.

The back-end consists of an Administrator application and a Lookup Engine. A unified schema represented in a Document Type Definition (DTD), and a repository of mappings are created to map from the data sources to the unified schema. Then, the Lookup Engine uses the repository of mappings to extract the relevant data from the data sources and create the unified data. The unified data is represented in XML and complies to the unified schema DTD. All the complexity of accessing the data sources, retrieving the data, and correlating it is done by the Lookup Engine, and is transparent to the application which just uses the output of the Lookup Engine - the unified data.

In 1999 we defined a Unified Schema DTD for performance, called performanceML, and it was published externally in xml.org.

The following figure sketches the proposed system:

(Embedded image moved to file: pic31736.jpg)

The front-end includes a Java visualization tool for rapid interpretation of temporal categorical data and is based on Plaisant & Shneiderman LifeLines. It allows you to see information at a glance while preserving the ability to drill down to see detailed backup information. It presents the data in layers and allows you to discover patterns. When there is a need for viewing and categorizing a lot of data, traditional human-interaction techniques may fall short. Long lists to scroll, clumsy searches, endless menus and lengthy dialogs will lead to user rejection. LifeLines include techniques to summarize, filter and present large amounts of information, leading us to believe that rapid access to needed data is possible with careful design.

The front-end includes the following features:

Each entity i.e. probe or server has its own shape (screen representation). We correlate one to the other by overlaying the servers shape over the probes shape.

The use of a colormap to highlight certain features of a presentation is a common technique in scientific visualization and image processing. Most often colormaps are used to apply color based on the

value of a continuous variable, however the use of categorical colormaps has been studied as well and is used here. It allows to direct attention immediately to the items requiring immediate attention.

Positioning on the Y axis is accomplished principally through categorization and sorting.

It includes a thumbnail overview and scrolling of the larger central visual representation. The thumbnail serves to direct the user's attention to appropriate parts of the display. The small rectangular box inside the thumbnail sketch performs the function of scroll bars.

It allows to drill-down to detailed information.

Deliverables (list the deliverables and describe for each what the value and impact will be...what will Global Services do with each deliverable):

An enhanced administrator application to map from data sources to the unified schema DTD.

impact: faster and easier creation of mappings by a visual application.

An enhanced DIX back-end system including a Lookup Engine that integrates the diverse data sources and creates the unified data. Performance and scaleability should be improved.

impact: generic infrastructure to ease data-integration problems.

An enhanced visualization applet that will report and analyze the unified data. Performance and scaleability should be improved.

impact: leverage system management services.

Ongoing evaluation on report requirements and incorporation to our report applet.

impact: leverage the quality of our report.

The deliverables of the two new directions will be specified if IGS is interested.

Technical (what will be done technically and why is research doing it...new technical ground or is it proving something in the marketplace):

The technology proposed for the back-end is new in the marketplace. A related technology is Virtual DB from Enterworks, but this proposal goes beyond Virtual DB by taking advantage of the new emerging XML technology. The technology proposed for the front-end was used before in the healthcare domain, and is new to the system performance domain. This will need new technical ground.

The technical work to be executed by HRL:

Enhance the DIX Look-Up Engine for the back-end and improve performance and scaleability.

Enhance the Administrator .

Implement new conversion functions for the new data sources.

Enhance the visualization applet and improve performance and scaleability.

The technical work to be executed by IGS:

Assistance in establishing connection to the new data sources.

Assistance in defining requirements/specifications.

Assistance in design review (optional).

Assistance in deploying in Raleigh.

Owner (who will take long term responsibility for this work, and what is their involvement now):

IGS Outsourcing

Competitive Analysis (who else, inside as well as outside IBM has solved or is solving this problem and what are the strengths and weaknesses of this solution):
Traditional solutions to integrating multiple data sources are of two types. One type is to perform the data integration at the application level (2-tier solution). This solution couples the business logic with the data and makes the application development expensive and difficult to support, customize or change. The other type of data integration is to create a new data warehouse and copy the data from the original diverse data sources to the warehouse. This solution is heavy and not flexible to dynamic changes in the data sources. IBM DataJoiner is a product that employs this solution.

Development Plan...

Tasks (list all of the project tasks as well as the Research and Global Services person months for each):

Staffing (Who are the people performing the tasks, and what is their availability - 50%, 100% etc.)

Simona Cohen - 100%
TBD - 100%
TBD - 100%
TBD - 100%

Detailed Milestones (List all of the milestones for this project - at least one per month):

Q1
Publish UCR technology in a referred conference.
Study the new data sources and establish connection to them.
Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).
DOU document.
Requirements document.

Q2
Design document.
Integrate SAP metrics data source with SRM. This will strive enhancements to the current components as well as development of new conversion functions.
Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).

Q3
Integrate new EPP metrics with SRM. This will strive enhancements to the current components as well as development of new conversion functions.
Add NLS to SRM by adding a new transformation that uses the IBM Automatic Translation Engine (optional).
Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).

Q4
Integrate Tivoli notes metrics with SRM. This will strive enhancements to the current components as well as development of new conversion functions.
Study how the system performance reports are used by the end-customer in order to improve the quality of the reports (ongoing).
Add documentation and code turn-over.

The milestons of the two new directions will be specified if IGS is

interested.

Support, Customization and Replication (How will this work continue to be executed once the project is complete):

Both the back-end and the front-end can be applied to other areas.

Development Environment (HW/SW used and where):

Development will be Java based and will use:

- Java development environment
- Windows/NT machines (400Mhz and above)
- AIX box
- DB2 for Windows NT and DB2 Connect
- Apache Web Server
- WebSphere Application Server

Financial Plan (Convert all PYs into $/quarter for Research, Global Services and total):

	RES	GS	TOT
1Q	$40K	$130K	$170K
2Q	$40K	$130K	$170K
3Q	$40K	$130K	$170K
4Q	$40K	$130K	$170K
TOT	$160K	$520K	$680K

ROI Overview:

Here is an estimation from 2000:

Project Initiated	1st Deliverable	First Year	Second Year	Third Year	Fourth Year	Each Year Thereafter
Time	Time 0 minus what (duration between start and 1st deliverable)?	$6.6M	$6.6M	$6.6M	$6.6M	$6.6M

Assumptions:

We estimate we spend $500k -$700k out of pocket per month to prepare customer reports -- the biggest expense was Kinkos.
The "effort" associated with gathering the data and preparing these

reports is significant but and we estimate this workload component to be 5 person yrs across all MO accounts.
Therefore, based on the above two assumptions, the annual cost savings would be approximately: ($500k*12) + ($120k*5) = $6.6M
There's also an intangible factor in that a solution here would demonstrate superior capability to a potential future customer and perhaps help convince a client to sign with IGS versus a competitor.

Here is SRM Interlock and savings/cost avoidance from 2000:

SRM needs to commit to cost avoidance and savings. The SRM development team commits up to 0.2 FTE savings (based on a small core development base to begin with) upon successful integration with SRM; we also commit up to 1 PM development cost avoidance savings for each new data metric that SRM will add in the future, such as Middleware metrics, etc... The SRM team will also contribute the necessary project management, architecture, and SRM (GUI, database) development hours in order to integrate the UCR solution with SRM 4.6 scheduled for delivery January/2001.

Customer expected cost savings needed at pilot completion (pursuing WW NCO account). Any customer savings, customer requirements, and feedback will be coordinated through Johnny Jones' NCO Project Office team. The current worldwide NCO install base which incorporates both SRM and EPP deliverables today, will be a natural fit for demonstrating UCR in a production environment. Follow-on demonstrations and team meetings will refine the customer requirements and expectations.

Technical pre-reqs The following technical pre-reqs documented in the UCR Plan will be executed by the SRM and IGS teams:

Assistance in establishing connection to SRM and EPP databases.
Assistance in understanding the new SRM and EPP schemas.
Assistance in defining requirements/specifications.
Assistance in design review (optional).
Selecting the pilot environment and customer.
Assistance in deploying the pilot in Raleigh.

ROI Details (Logic behind estimate like -- Helpdesk agents receive repetitive calls on common problems that could easily be resolved by the customer if they had the correct procedure to provide educational instructions. These calls account for approximately 25-33% of the total volume received by the help desk. Putting this into perspective by applying it to South SDC call volumes (i.e. 2M per year), this correlates to about 500K-660K calls per year. Assuming a minimal 25% success factor on these calls, a rough resource savings would be 25 FTE(@ 500 calls per FTE/month) for the help desk):

Presentations (suitable for use with senior management to present all of the information described above - revise the status charts, as well as the status section above, on a monthly basis):

APPENDIX C

Simona Cohen on 19/07/2000 at 08:37
Category: IGS Meeting Minutes
(Embedded image moved to file: pic22925.jpg)

Attendees:
Keith Kempke/Boulder/IBM@IBMUS
Amy Hahn/Lexington/IBM@IBMUS
Johnny Jones/Raleigh/IBM@IBMUS
Simona Cohen/IBMHAIFA@IBMHAIFA

Minutes:
Jeff, Gary and Andy are on vacation.
We have now an SRM test environment installed in Haifa.
Keith is missing some information about the replication probes to DBNotes servers. Then, he will enter those additional probes to the Boulder database.
We saw some of the reports in Keith web site at
http://wrtdata.boulder.ibm.com/rtm/login.asp
user - srm
password - ibmcorp
They were created with Chart-FX reporting tool, and next they will use BRIO.
The site is impressive!

Simona Cohen on 19/07/2000 at 15:46
Category: Meeting Minutes
(Embedded image moved to file: pic23290.jpg)

Attendees:
Noga Meshulam/IBMHAIFA@IBMHAIFA
Leonid Dubinsky/Haifa/IBM@IBMHAIFA
Simona Cohen/IBMHAIFA@IBMHAIFA

Minutes:
SRM site is installed in Haifa, and we have the code to connect our staff into it.
Noga is adding the use of templates to the adminTool.
The Lookup Engine knows to do a JDBC2XML and XSLT transformations for SRM database.
We went through the applet and discussed the "more info" window, adding focus on the item selected, adding hourly probes for the applet with large data, checking the data.

APPENDIX D

Simona Cohen on 26/07/2000 at 15:12
Category: IGS Meeting Minutes
(Embedded image moved to file: pic01473.jpg)

Attendees:
Jeffrey R Baker/Watson/IBM@IBMUS
Gary Quesenberry/Raleigh/IBM@IBMUS
Keith Kempke/Boulder/IBM@IBMUS
Amy Hahn/Lexington/IBM@IBMUS
Johnny Jones/Raleigh/IBM@IBMUS
Simona Cohen/IBMHAIFA@IBMHAIFA

Minutes:
Keith is checking the inconsistency in the data found in the Boulder summary tables, and will send us a note when this is resolved.
Keith is working on inserting the new ReplicProbe to Boulder, and will send us a note when this is done.
We'll have a new demo by the end of the month with real data. We'll notify all when this is ready, and then Gary will schedule a call to gather feedback.
Gary will put in the SRM next version plan an item to integrate UCR into SRM. SRM next version is scheduled for October.

38070
תיק 1819

IBM

APPENDIX E
IBM

Phone:
FAX: 04-855-0070
email:

Facsimile

38070

To: דניאל קיגלר

@Fax: 06-6456536

From: סימונה כהן

Date: 3/8/2000

Re:

Pages: 7, including this

דניאל,

אני שולחת את החומרים לגבי המפגש השני.

לגבי כל בעיה אתה יכול להתקשר אלי:

בבית – 04-8340925

בעבודה – 04-8296479

תודה רבה,
סימונה כהן
IBM

INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE

FIELD OF THE INVENTION

The present invention relates generally to information systems, and specifically to methods and systems for integration of heterogeneous data from distributed sources.

BACKGROUND OF THE INVENTION

In today's business environment, many applications need to use data that are warehoused in diverse data sources and repositories. These data are typically expressed in different formats and languages, and are retrieved using different access methods and delivery vehicles. Intermittently, new data sources may be added to the corpus that the application must handle, while existing data sources may be removed or changed. These problems are particularly prevalent in organizations in which databases and applications have developed gradually over the course of years in response to growing Information Systems (IS) demands.

There is therefore a need for tools that enable database information from diverse sources to be integrated into an accessible whole. Such tools are required, for example, in such application areas as customer relationship management, personnel data warehouses, and performance analysis of computer systems and networks. The conventional approach to meeting this need is to create a new data warehouse and to copy into it the required data from the original sources. For example, IBM Corporation (Armonk, New York) offers a product known as "DB2 DataJoiner" that is based on this

which specifies the elements that can exist in the document and the attributes and hierarchy of the elements. Many different DTDs have already been developed for different domains ~~applications~~, such as "performanceML", domain for computer system performance evaluation, and "CPEX" domain for customer relationship management. XML.ORG maintains a registry of available DTDs at xml.org/xmlorg_registry/index.shtml. XML-schema is ~~are~~ under development as an alternative to DTD~~s~~, as described at www.w3.org/TR/xmlschema-0.

Style languages are used to control how the data contained in a markup language document are structured, formatted and presented. For example, W3C has introduced the Extensible Style Language (XSL) for use in defining style sheets for XML documents. An XSL style sheet is a collection of rules, known as templates. When the rules are applied to an input XML file by a processor running an XSL engine, they generate as output some or all of the content of the XML file in a form that is specified by the rules. (In fact, an XSL style sheet is itself a type of XML document.) XSL includes a transformation language, XSLT, which is defined by a standard available at www.w3.org/TR/xslt. Rules written in XSLT specify how one XML document is to be transformed into another XML document. The transformed document may use the same markup tags and DTD as the original document, or it may have a different set of tags, such as HTML tags. Other style languages are also known in the art, such as the Document Style, Semantics and Specification Language (DSSSL), which is commonly used in conjunction with SGML.

hereinbelow. The schema is preferably specified by a DTD that is defined for the domain to which data sources 32 belong, such as the performanceML or CPEX DTD noted above. The mappings defined by administrator 34 are stored in a repository 36.

A lookup engine 40 uses the mappings in repository 36 to access data sources 32 and to map the diverse data from these sources to unified data 42. The unified data are preferably represented as XML code, complying with schema 38. An application such as a query engine 44, typically running on client 22, is able to access unified data 42 substantially without regard to the differences in format and access methods among sources 32. From the point of view of the application, the diverse sources are all a single body of XML data. Query engine 44 preferably comprises an XML-based query engine, written in the XQL or XML-QL query language, for example. **(Can you provide a reference describing these query languages or query engines?)** see below

many times the query engine is in the server side and only parameters come from the client

Fig. 3 is a flow chart that schematically illustrates a method by which system 20 integrates the data from sources 32, in accordance with a preferred embodiment of the present invention. At a schema creation step 50, unified schema 38 for the selected domain is specified. Preferably, an existing DTD is selected. Alternatively, a new DTD or other schema may be created, or another type of schema may be used, such as an XML-schema, as mentioned above. The schema should be adequate to cover all of the existing types of data in the domain and, preferably, should be easily extensible

XQL - http://www.w3.org/TandS/QL/QL98/pp/xql.html

XML-QL - http://www.w3.org/TR/1998/NOTE-xml-ql-19980819

to allow the addition of tags defining new data elements within its hierarchy.

At an access step 52, administrator 34 defines how each of data sources 32 is to be accessed. For example, if the data source is a database, the administrator preferably defines a host name and port (on a network linking server 24 to the appropriate storage device 26), the database name, and a username and password. The administrator then creates mappings from data sources 32 to schema 38, at a mapping step 54. Preferably, the mappings are created using an on-screen editor, as illustrated in Fig. 4. The editor preferably creates XSLT rules, which are used subsequently to carry out the actual mapping. Alternatively, a user of administrator 34 may create the XSLT rules (or other mapping function) by coding it directly, without the aid of a visual editor.

Preferably, each of the mappings created at step 52 is a triplet of the form <source, target, conversion function>. The source is a field or a set of fields in data source 32. The target is an element or an attribute, or a set of elements or attribute in unified schema 38. The conversion function is a function that is applied to the data in the source in order to create a data value for the target. For example, assuming that system 20 is assembling computer performance data using the performanceML DTD, one of the triplets might be as follows:

- Source – Day.cpu_utilization (the CPU utilization field in the computer's Day table).

5. Select certain rows in the data source, apply a conversion function to each selected row, and create an element or attribute in the DTD.
6. Join tables in the data sources to one element in the DTD. The tables may belong to different data sources. **{What does it mean to "join tables"?}** combining rows from several tabl
7. Aggregate data from the data source with a simple function - for example, find the average response time per day. Copy the aggregated data to an element or attribute in the DTD.
8. Aggregate data from the data source with a complex conversion function. Copy the aggregated data to an element or attribute in the DTD. **{What is the definition of "complex" in this context, as opposed to "simple"?}** a function that you need to write some code to implement and is not given by the XSLT.
9. Mappings that include parameters - for example, copy column cpu_utilization_objective to $cpu_objective, wherein the parameter cpu_objective can get different values in each execution of lookup engine 40.

As noted above, the mappings created at step 54 are preferably recorded as XSLT rules. Tables II and III below are examples of XSLT code that implements two rules of this sort. Table II is a mapping of the first type (direct copy) listed in Table I, while Table III is a mapping of the ~~fifth~~ eighth type (~~row conversion~~). (aggregate with a comple conversion function)

TABLE II - XSLT DIRECT COPY

source - 'fqhn' column of table 'perform.server'
target - attribute 'Server_id' of element 'Server_Configuration_Info' in DTD

14. Apparatus according to claim 13, wherein the unified schema comprises a Document Type Definition (DTD).

15. Apparatus according to claim 13, wherein the definitions of the correspondences comprise data transformation rules in Extensible Style Language (XSL).

16. Apparatus according to claim 15, wherein the processor is adapted to map the source data by transforming the data using an XSL engine.

17. Apparatus according to claim 12, wherein each of the definitions of the correspondences comprise a selection of one or more of the data fields in the sources to correspond to one or more of the markup tags in the schema, together with a conversion function to be applied by the processor to the one or more source fields.

18. Apparatus according to claim 12, wherein at least some of the source data are represented in a language other than the markup language, and wherein the processor is adapted to transform the data to the markup language.

19. Apparatus according to claim 12, wherein the processor is adapted to receive and respond to a query addressed to the unified data in the markup language.

20. Apparatus according to claim 19, wherein the processor is adapted to map the source data responsive to the query.

21. Apparatus according to claim 12, and comprising a plurality of distributed data storage devices, which hold the diverse data sources, wherein the processor is adapted to retrieve the source data from the distributed devices.

APPENDIX F

38070
ז'ארל
1819

STC

Sanford T. Colb & Co.
Intellectual Property Law

Beit Amot Mishpat
8 Shaul Hamelech Blvd.
Tel-Aviv 64733, Israel
Tel. 972-3-693-8560

4 Shaar Hagai
P.O. Box 2273
Rehovot 76122, Israel
Tel. 972-8-945-5122

Beit Lev Hagivah
11 Beit Hadfus
Jerusalem 95483, Israel
Tel. 972-2-651-9453

Facsimile: 972-8-945-4556 972-8-949-1040 ◆ e-mail: colbpat@stc.co.il

IBM CONFIDENTIAL
23 pages via fax to 04-8550070

August 5, 2000

Ms. Simona Cohen
IBM ISRAEL
Haifa Research Laboratory
MATAM, Haifa 31905

Re: New U.S. patent application
INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE
Your ref. IL9-2000-0027, our ref. 38070

Dear Simona:

Attached please find a revised draft of the above-referenced patent application incorporating your comments and corrections. In preparation for filing, I have also added an abstract and paraphrased the claims in the Summary of the Invention. As the figures are unchanged, I am not sending them again.

Please review this draft, together with your co-inventors, and let me have your approval to file the application. If the application is acceptable as it stands, please ask Tal Noy-Cohen to send me her authorization by fax or e-mail to file the application. I must have Tal's authorization no later than tomorrow (Sunday) evening in order to assure that the application is filed on Monday, before I leave on vacation.

Sincerely yours,

Daniel Kligler, Ph.D.
Sanford T. Colb & Co.

encl.

cc: Adv. Tal Noy-Cohen

38070
12/9

From: noy@il.ibm.com
Sent: Monday, August 07, 2000 9:47 AM
To: dkligler@stc.co.il
Subject: Approval to file

Daniel, please file the following cases: 36694, 37590, 38071, 38070, 39272

Thanks and enjoy your vacation,
Tal

Tal Noy-Cohen, adv.
Intellectual Property Department - Operation Services, IBM HRL
Voice: +972-4-829-6274 Fax: +972-4-829-6521 mail: noy@il.ibm.com

APPENDIX H

סנפורד ט. קולב ושות', עורכי־דין ועורכי פטנטים

SANFORD T. COLB & CO., ADVOCATES & PATENT ATTORNEYS

SANFORD T. COLB סנפורד ט. קולב
LL. B. (CANTAB), B.A., M. Sc. (PHYSICS), J. D. (HARVARD)
RUTH SEGEL LL. B. (LONDON) רות סגל
EITAN SHAULSKY LL. B. איתן שאולסקי
YISRAEL SAPERSTEIN LL. B. ישראל ספירשטיין
MIRIT ELDOR LL. B. מירית אלדור
MICHA KAUFMAN LL. B. מיכה קאופמן

COUNSEL: יועץ:
REUVEN BOROKOVSKY ראובן בורוקובסקי
LL. B., B. A. (ECON. & STAT.), M. B. A.
MICHAEL OPHIR LL. B. מיכאל אופיר

משרד רחובות: ת. ד. 2273, רחובות 76122, טל. 08-9455122, פקס. 08-9491040, 08-9454556
משרד תל־אביב: בית אמות משפט, שד' שאול המלך 8, תל-אביב 64733, טל. 03-6938560, פקס. 6938561
משרד ירושלים: לב הגבעה, רחוב בית הדפוס 11, ירושלים 95483, טל. 02-6519453, פקס. 02-6519454
משרד חיפה: בית טופז, קומה שלישית, שער הכרמל, חיפה, טל. 04-8503444, פקס. 04-8503555
REHOVOT OFFICE: P.O.B. 2273, REHOVOT, TEL. 08-9455127, TELECOPIER 08-9454556, 08-9491040
TEL-AVIV OFFICE: BEIT AMOT MISHPAT, 8 SHAUL HAMELECH BOULEVARD 64733 TEL-AVIV, ISRAEL, TEL. 03-6938560, TELECOPIER 03-6938561
JERUSALEM OFFICE: LEV HAGIVAH, 11 BEIT HADFUS, JERUSALEM, TEL. 02-6519453, TELECOPIER 02-6519454
HAIFA OFFICE: BEIT TOPAZ, 3RD FLOOR, SHAAR HACARMEL, HAIFA, TEL. 04-8503444, TELECOPIER 04-8503555

E-Mail: colbpat@stc.co.il

August 9, 2000

BY SPECIAL MAIL DELIVERY

Adv. Tal Noy-Cohen
IBM ISRAEL
Haifa Research Laboratory
Matam
Haifa 31905

Re: New U.S. Patent Application
INTEGRATING DIVERSE DATA SOURCES USING A MARK-UP LANGUAGE
Your Ref.: IL9-2000-0027
Our Ref.: 38070

Dear Ms. Noy-Cohen,

Enclosed please find two copies of the specification, claims and drawings of your new U.S. application. Attached to the back of one copy are the Declaration and Power of Attorney and Assignment Document.

Please have the Declaration and Assignment forms duly signed by the inventors, Simona Cohen, Tirtsa Hochberg, Haim Nelken, Ilan Paleiov and Pnina Vortman, where indicated and return the specification, **with the documents still attached**, to our Rehovot office as soon as possible.

The other copy of the specification, claims and drawings is for your file.

Sincerely yours,

Einat Niv
Foreign Filing Department
Sanford T. Colb & Co.